ECE 792 - Advanced Machine Learning
PROJECT FINAL SUBMISSION REPORT (GROUP-1)

Team Members:
Aman Waoo - 200483401
Girish Wangikar - 200475452

Option D: Project-Oriented
Topic: Demonstration of Model for Deepfake Detection

1. Problem Statement

The rise of deepfake technology has led to the creation of fake images and videos that are nearly impossible to distinguish from real ones. For instance, they can be used to synthesize fake face images in pornography videos or speech videos with synthesized facial content of famous politicians, causing severe problems for society, political, and commercial activities. This technology poses a significant threat to individuals who may be targeted by it, highlighting the urgent need for effective algorithms to detect and differentiate real media from deepfakes. Traditional image forgery detection approaches use active and passive schemes, but these methods are not effective for Generative Adversarial Network (GAN) generated fake images. Therefore, deep neural networks have been adopted to detect fake images generated by GANs, treating fake image detection as a binary classification problem. However, these approaches are limited to detecting partial manipulation of face images and cannot detect fully generated fake images. In addition, supervised learning strategies tend to overfit the training data and may not have good detection ability. Thus, deepfake face detection remains a complex and challenging task for humans to accomplish without assistance. Current research in generating and detecting deepfakes has produced various GAN architectures that utilize convolutional neural networks (CNNs) to create realistic fake images. However, detecting deepfakes across multiple GAN architectures by generalization still remains a challenge, and many existing detection models do not utilize statistics to improve detection accuracy.

2. Objectives

- To explore the problem of detecting face images generated by various GAN architectures, which is currently a challenging task.
- To address the limitations of existing detection models, which include the inability to generalize detection across untrained GAN architectures and the lack of statistical analysis between real and fake images.
- To propose a modified version of the pairwise learning model introduced by Hsu C. et al. [1], that can effectively detect fake face images generated by different GAN architectures.
- To evaluate the performance of the proposed model and compare it with existing detection models through various metrics such as accuracy, precision, recall, and F1 score.
- To visualize the network architecture and learn the model's learning process for better understanding and interpretation.

3. Model Architecture

The model architecture contains two separate blocks, a novel common fake feature network (CFFN) to extract the discriminative common fake features (CFFs) between image pairs via

contrastive learning, and the classification network (CN) that utilizes the outputs from the trained CFFN to make a decision on the incoming image as real or fake.

3.1. Common Fake Feature Network (CFFN)

While advanced CNNs like DenseNet, ResNet, and Xception can be used to learn fake features from the training set, they are mostly trained in a supervised way. To address this, the project proposes integrating the Siamese network with DenseNet to develop CFFN for discriminative CFF learning. CFFN is a two-streamed network that allows pairwise input for CFF learning, making it possible to learn common features that are difficult to learn using traditional CNNs.

The CFFN consists of four dense units, including two, three, four, and two dense blocks, respectively, and the number of channels in the four dense units are 48, 60, 78, and 126, respectively. The parameter θ in the transition layer is 0.5, and the growth rate is 24. In addition, the cross-layer features are integrated into the classification layer to improve the fake image recognition performance.

The convolution and fully connected layers are concatenated into the last convolution layer of the proposed CFFN to obtain the final concatenated feature vector. The method involves two-step learning that combines the CFF based on pairwise learning and the classification network based on the typical binary cross-entropy learning method. Thus, The CFF is a semi-supervised construct that allows for arbitrary fake and real images for training and simply requires a binary indicator showing if a paired image to the network is real-real or fake-fake.

During the training of the CFFN, the outputs from the last convolutional layer and the last two dense convolution blocks are concatenated to a fully connected network that contains the discriminative features of the input image. The architecture of the CFFN designed in the project includes six layers, with layer one being a convolutional layer with a kernel of 7x7, stride=4, and 48 filters. Layer two is a dense block with two hidden convolutional layers and 48 filters, while layer three is a dense block with three hidden convolutional layers and 60 filters. Layer four is a dense block with four hidden convolutional layers and 78 filters, layer five is a dense block with two hidden convolutional layers and 126 filters, layer six is a convolutional layer with a kernel of 3x3 and 128 channels, and layer seven is a fully connected layer with 128 neurons. The discriminative feature representation is obtained by adding the fully connected layer. As a result, the final feature representation has 384 neurons (128 × 3).

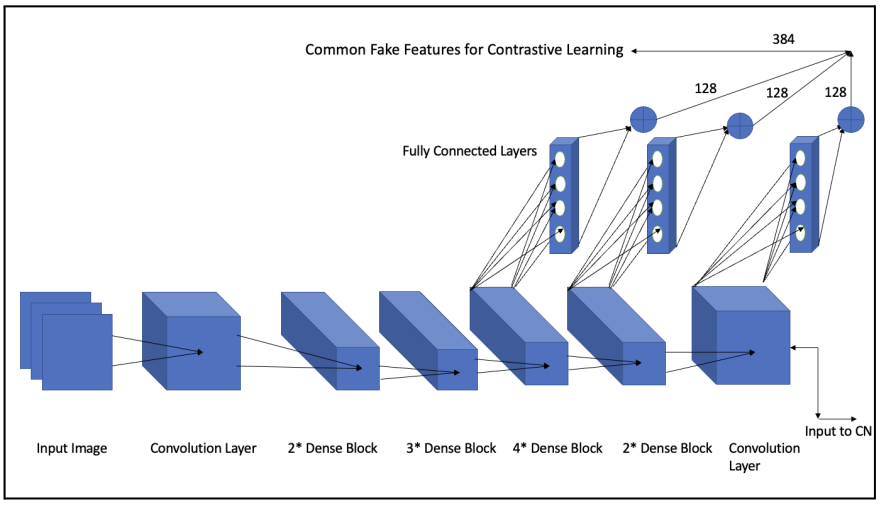

Fig. 1. CFFN architecture

3.2. Classification Network (CN)

The classification network architecture, as explained in [1], is used to classify detected images into two categories: real and fake. To enhance the performance of the classification network, a sub-network is implemented as a classifier. The sub-network is composed of a convolution layer with two channels and a fully connected layer with two neurons, again, as stated in [1]. The classification process is achieved by using the cross-entropy loss function, and the classifier is trained using the back-propagation algorithm. The proposed learning method consists of two steps: pairwise learning and classifier learning. The discriminative common fake feature (CFF) is learned using contrastive loss through the proposed CFFN. Once the CFF is learned, the classification network captures it to identify whether the image is real or fake.

Compared to the CFFN, the classification network is much simpler as most of the classification work is expected to be done by the optimally trained CFFN. The CN is responsible for projecting the CFFN's high-dimensional output into a 2D vector representation, which can be used to perform binary cross-entropy loss (BCE) to determine if an image is real or fake. The CN has a convolutional layer with a 3x3 kernel and two output channels. The output is then average-pooled and fed into an Fully Connected Network (FCN), where the final 2D vector output is used to calculate the loss.

During training, only one image is passed through the entire network at a time, unlike the CFFN, where a pair of images is used at each training step. Fig. 2 shows the architecture of the CN, where the CFFN's output is labeled as the “Input to CN” in Fig. 1. This approach addresses the difficulty of collecting training samples generated by all possible GANs and eliminates the need to retrain the fake face detector for every new GAN.

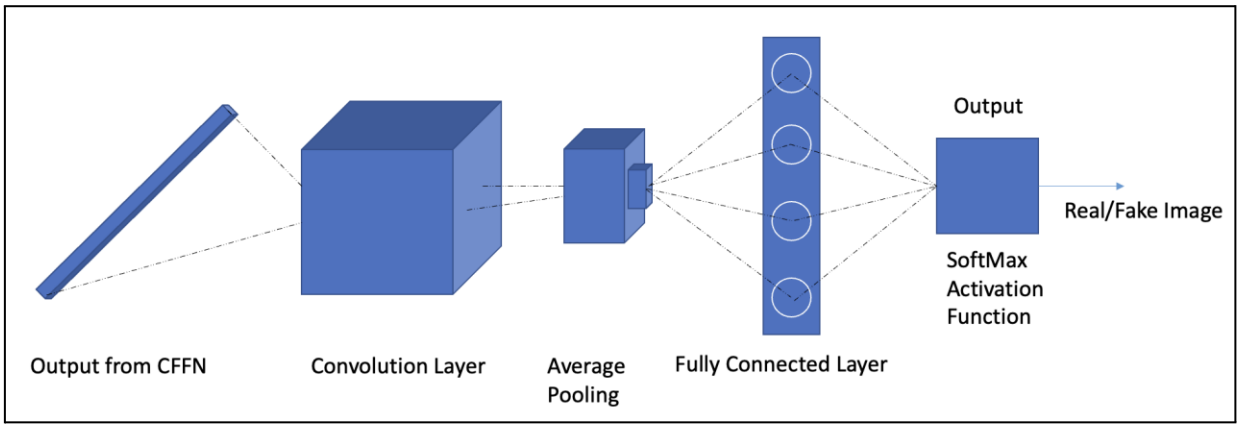

Fig. 2. CN architecture

Layer Number	Feature Learning	Classification
1	conv. layer, kernel = 7 × 7, stride = 4, #channels = 48	Conv. layer, kernel = 3 × 3, #channels = 2
2	Dense block × 2, #channels = 48	Global average pooling
3	Dense block × 3, #channels = 60	Fully connected layer, neurons = 2
4	Dense block × 4, #channels = 78	
5	Dense block × 2, #channels = 126	
6	Fully connected layer, neurons = 128	SoftMax layer

Table 1. Parameters of the proposed CFFN and CN models

4. Network Learning Policies

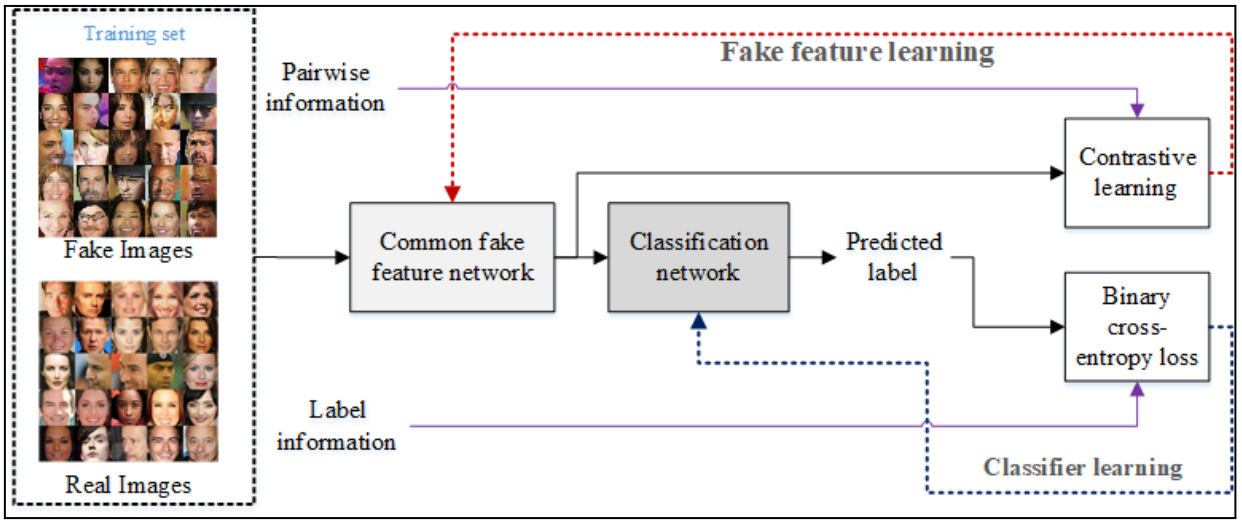

Fig. 3. Flow chart depicting the proposed fake face detector with the learning strategies

4.1. Discriminative Feature Learning

One of the main limitations of supervised learning is the difficulty in recognizing the excluded subjects during the training phase. To address this issue, contrastive loss is incorporated into the energy function of the cross-entropy loss to learn discriminative common fake features (CFFs) by pairwise learning. To facilitate pairwise inputs, the Siamese network structure is employed. During the training process of the proposed CFFN, the contrastive loss term is utilized to learn discriminative features.

Given a face image pair (x_1, x_2) and the pairwise label y, where $y = 0$ indicates a fake pair, and $y = 1$ indicates a genuine pair, the energy function between the image pair is defined as,

$$E_W(x_1, x_2) = \|f_{CFFN}(x_1) - f_{CFFN}(x_2)\|_2^2$$

where, f_{CFFN} represents the 384-length feature vector for the image pair (x_1, x_2).

The aim is to minimize the energy function $E_W(x_1, x_2)$; however, direct computation of energy function by calculating l_2 norm distance in the feature domain leads to a constant mapping, where any input is mapped to a constant vector, resulting in a useless feature representation.

Therefore, to overcome this problem, the contrastive loss function can be expressed as,

$$L(W, (P, x_1, x_2)) = 0.5 \times \left(y_{ij} E_w^2\right) + \left(1 - y_{ij}\right) \times \left(0, \left(m - E_w\right)_2^2\right)$$

where, m denotes the predefined threshold value, in our case, equals to 0.5. When $y_{ij} = 1$ (i.e., for a genuine pair), the cost function tends to minimize the energy (defined by the feature distance) E_W between two images. Conversely, when $y_{ij} = 0$ (i.e., for an impostor pair), the contrastive loss function tends to minimize the function $max(0, \left(m - E_w\right))$. This means that the energy E_W will be maximized if the feature distance between the impostor pair is smaller than the predefined threshold value m. The contrastive loss enables the feature representation $f_{CFFN}(x_i)$ to become similar to $f_{CFFN}(x_j)$ when y_{ij} = 1, which helps to learn the common characteristics of fake images generated by different GANs. By iteratively training the network f_{CFFN} using the contrastive loss, the CFFs of the collected GANs can be learned effectively.

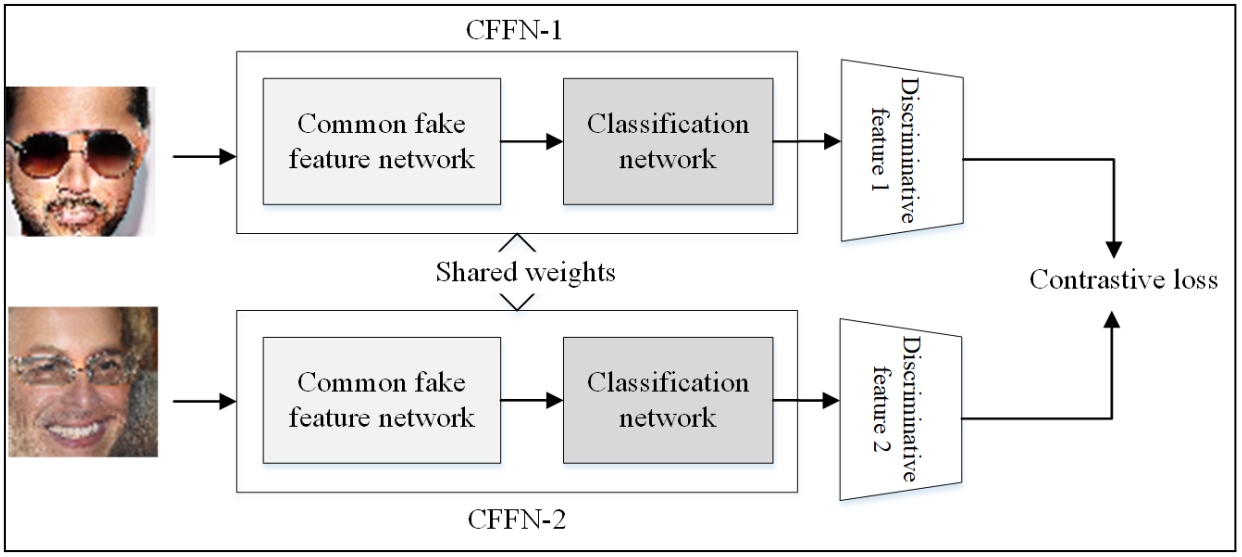

Fig. 4. Discriminative feature learning approach in proposed network

4.2. Classification Learning

In order to improve the performance of fake image detection, a sub-network is utilized as a classifier. The classification learning is achieved by using the cross-entropy loss function and subsequently, back-propagation algorithm can be used for training the classifier. The loss function for classification learning, is given as,

$$L_c(x_i, p_i) = -\sum(i, N_T)\left(f_{CLS}(f_{CFFN}(x_i))log(p_i)\right)$$

where, $p_i = 0$ indicates the real image, and f_{CLS} denotes the classification sub-network.

There are two strategies to learn both the CFFs and the classifier. The first strategy is to use joint learning, which incorporates the contrastive loss and cross-entropy loss functions into the total energy function. However, it can be difficult to observe the impact of both loss functions on the performance of fake image detection tasks with this approach. The second strategy is where the CFFN is first trained using the proposed contrastive loss, followed by training the classifier based on the cross-entropy loss.

4.3. Two-Step Learning Policy

The two-step learning policy, which is adopted in this project, involves two loss functions, namely contrastive loss and cross-entropy loss, for the CFFN and classifier learning, respectively. In order to prioritize the primary purpose of the proposed CFFN, which is to learn discriminative features, the CFF can be learned first by minimizing the contrastive loss. Once the CFF has been trained, any classifier can be used to recognize the fake face image based on the learned CFF. To enable end-to-end training and enhance classifier performance, a small neural network can be used as the classifier since it is well-known that a classifier can be easily trained based on better feature representation and optimized by minimizing the cross-entropy loss. To validate the effectiveness of the two-step learning policy, an experiment is conducted in [1] to compare its performance with the joint learning approach (i.e., Baseline-I) in the experimental section.

5. Experimental Results

5.1. Dataset Generation

The dataset used in the experiment was obtained from CelebA data [2], which provided a diverse range of images covering a large variety of poses and backgrounds, consisting of 10,177 identities and 202,599 aligned face images. To generate the fake image dataset, three

state-of-the-art GANs were utilized, including DCGAN [3], WGAN [4], and PGGAN [5].

5.1.1. Deep Convolutional GAN (DCGAN):

- Uses convolutional layers instead of fully connected layers to capture spatial features and generate more realistic images.
- Introduces architectural constraints like batch normalization and ReLU activation to stabilize training.
- Generates high-quality images with realistic details and textures.

5.1.2. Wasserstein GAN (WGAN):

- Aims to improve the stability and convergence of GAN training.
- Uses Wasserstein distance or Earth Mover's Distance (EMD) to measure the distance between the real and fake distributions.
- Replaces the binary output of the discriminator with a real-valued output that represents the estimated Wasserstein distance.

5.1.3. Progressive Growing GAN (PGGAN):

- Uses a progressive training strategy to gradually increase the resolution of generated images.
- Starts with a low-resolution generator and discriminator and gradually increases their depth and resolution over multiple training iterations.
- Uses several architectural innovations to capture fine-grained details and produce high-quality images with realistic textures and structures.
- Innovations include pixel-wise feature normalization, progressive growing of convolutional layers, and using skip connections between the generator and discriminator.

However, except for PGGAN, the GANs were unable to synthesize high-quality images with a high resolution of 128 × 128 pixels, and the fake images generated by the GANs were of a default size of only 64 × 64 pixels. In order to ensure that the performance of different image detectors could be compared fairly, the input image size was kept consistent at 64 × 64 pixels. We obtained pre-existing code from a github repository [3] to train the DCGAN model on the CelebA dataset. The code was designed to replicate the original DCGAN paper [6], and we trained the model for 50 epochs on the entire CelebA dataset, resulting in the generation of 40,000 fake images.
For the WGAN dataset, we used a pre-existing model trained on the CelebA dataset that was available at [4]. We generated 40,000 fake images using this pre-existing model.
To generate the PGGAN dataset, we loaded the publicly available PGGAN model from [5]. This model was originally trained on the high-quality CelebA dataset, although we resized the generated images to 64x64.
To summarize, we collected 120,000 fake images from three GAN models: DCGAN, WGAN, and PGGAN. Additionally, we included 120,000 real images from the CelebA dataset, which we selected to match the number of fake images generated. The inclusion of both real and fake images allows for a balanced dataset that can be used for supervised learning approaches. With this dataset, we can evaluate the effectiveness of our proposed method for fake image detection and compare it with existing approaches.

5.2. Model Training (CFFN + CN)

To train our models, we split our dataset into 80% training data and 20% testing data. We trained our Common Fake Feature Network (CFFN) for 15 epochs, using 2,000,000 image pairs of fake-fake and real-real combinations and a batch size of 88, as given in [1]. Figure 5

shows the training loss over the 15 epochs, indicating that the loss was continuously decreasing during training, suggesting that the model could continue to learn further. Additionally, we only used 2,000,000 image pairs, which did not allow us to utilize all the available 120,000 real and 120,000 fake images, as creating pairs for all these images would not be feasible. We also kept fake-fake image pairs with their own GAN type that generated them to learn the common features between similar images.

After training the CFFN, we trained the Classification Network (CN) using 25 epochs and a batch size of 88, as given in [1], utilizing all real and fake images. During CN training, the output from the last convolutional layer of the CFFN served as the input. Figure 7 shows that the loss of our CN was continuously decreasing when we stopped training, indicating the possibility of further training. The accuracy of our model was also improving, as shown in Figure 6, with an accuracy of 99.3% in correctly labeling real and fake images.

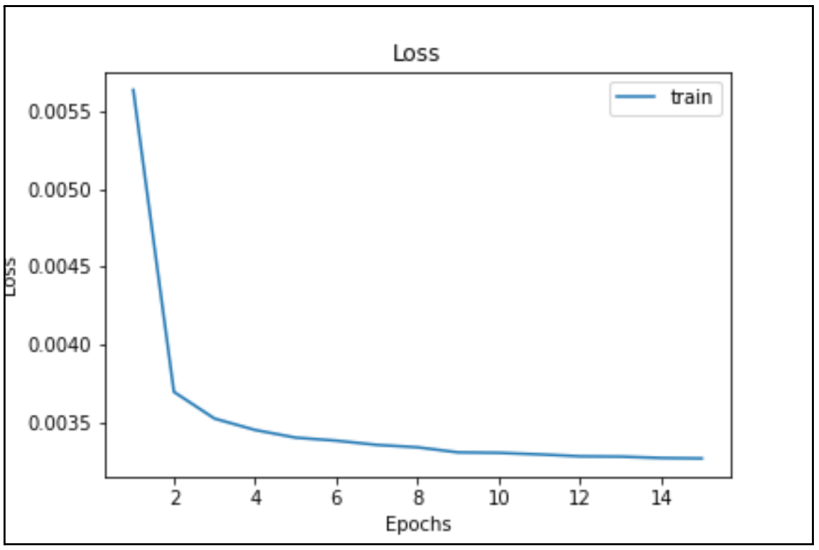

Fig. 5. CFFN Training Loss

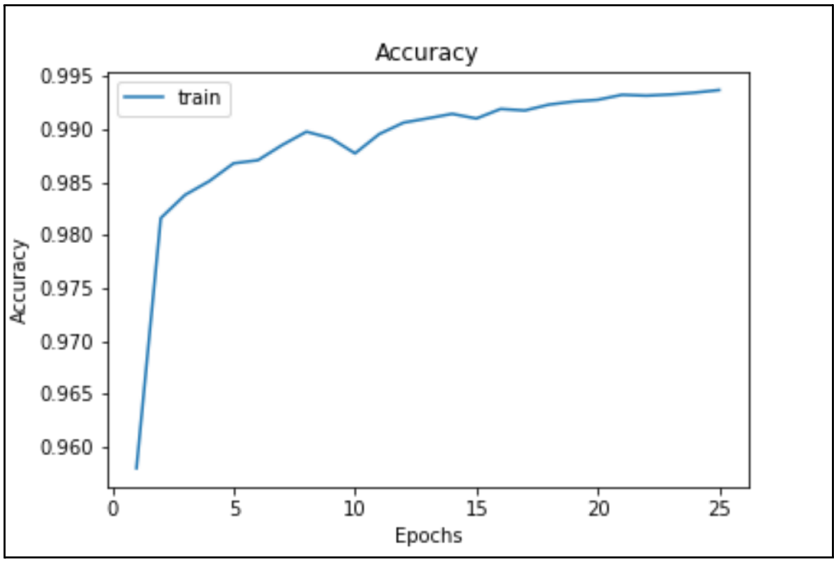

Fig. 6. CN Training Accuracy

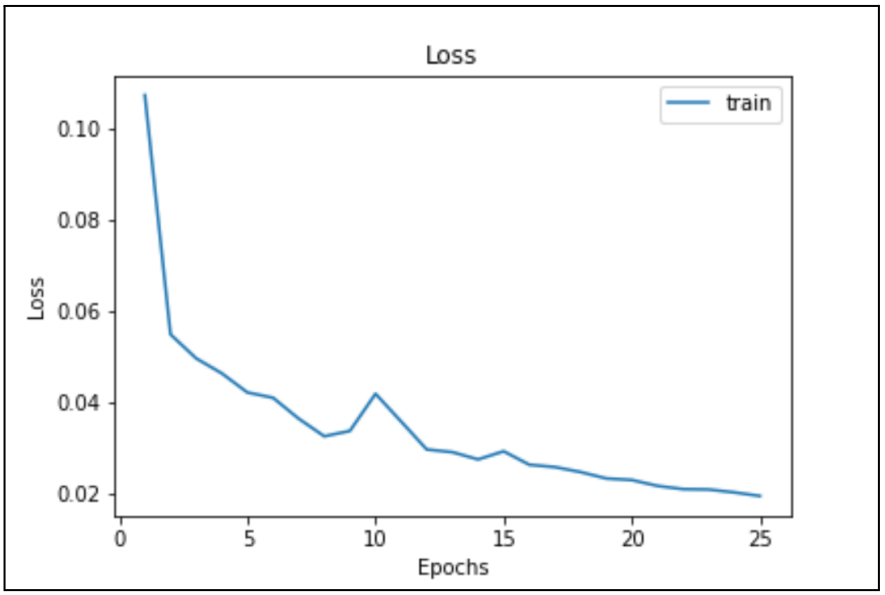

Fig. 7. CN Training Loss

5.3. Performance Comparison

Regarding the performance of our proposed model for fake face image detection, we conducted a comprehensive performance comparison with other existing methods, as described in [1]. We retained accuracy statistics for each GAN, and we utilized both our original fake and real datasets to evaluate our network's performance. The results are presented in Table 2, which compares the performance of each GAN against our model trained on 120k real images and 40k fake images generated by WGAN-CP, DCGAN, and PGGAN.

Method/Target	**DCGAN**		**WGAN-CP**		**PGGAN**	
	Precision	*Recall*	*Precision*	*Recall*	*Precision*	*Recall*
Baseline-I [1]	0.882	0.887	0.902	0.920	0.938	0.901
Baseline-II [1]	0.822	0.838	0.864	0.881	0.917	0.887
Proposed [1]	0.929	0.916	0.988	0.927	0.988	0.948
Our network	**0.942**	**0.948**	**0.991**	**0.984**	**0.989**	**0.978**

Table 2. Precision & Recall Values per GAN comparison with reference to Hsu C. et al. [1]

Our model's performance was better than other methods in terms of precision and recall, including the proposed method in [1], which we aimed to emulate. However, several factors could have contributed to this superiority. We implemented the dense blocks of the CFFN ourselves based on additional resources, which may have caused some discrepancies between our models. Additionally, our generated GAN images may not have been of high quality, which could have made them easy to differentiate.

To shed some additional light on the performance of our model, it would be enlightening to train our classification network with only the original real and fake images, rather than preprocessing the images through our CFFN, and compare the performance.

Despite these uncertainties, our model demonstrated high accuracy against GAN images that suggests that our model can generalize against different fake face GAN images. However, there are also concerns about the quality of our GAN dataset, as there are currently no standardized and reliable method for generating the desired fake GAN images.

5.4. Image Reconstruction using DeConvNet

One approach to analyze convolutional networks is through image reconstruction using the DeConvNet model, as followed in Homework 2. This involves backpropagating through the network to undo all data transforming operations, such as convolution, pooling, and concatenation, to obtain the filter activations. By visualizing these activations, we gain insight into the discriminative features learned by the network. In our project, we applied this technique to the CFFN to visualize the features that distinguish real images from fake ones generated by DCGAN, WGAN-CP, and PGGAN.

Our results in Fig. 8 - Fig. 11 showed that there are many filters that completely wipe away information in the GAN-generated images, indicating that these images lack distinguishable features. In contrast, the real images have more nonzero filters, suggesting that they contain more discernible features that can be used to identify them. It is also worth noting that the quality of GAN images can affect the network's detection capabilities. These filter activations are then fed into our classification network to differentiate between real and fake images.

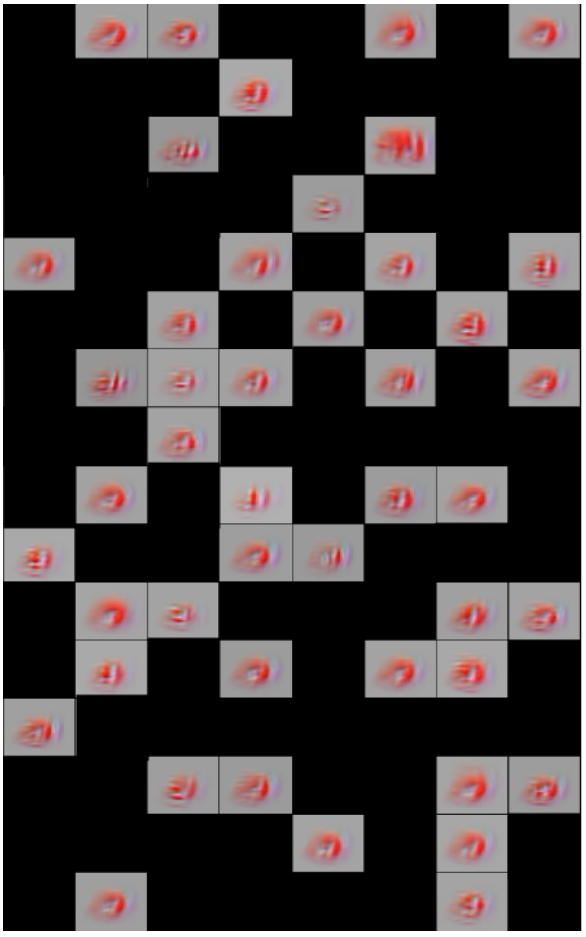

Fig. 8. Filter activation visualization for real image

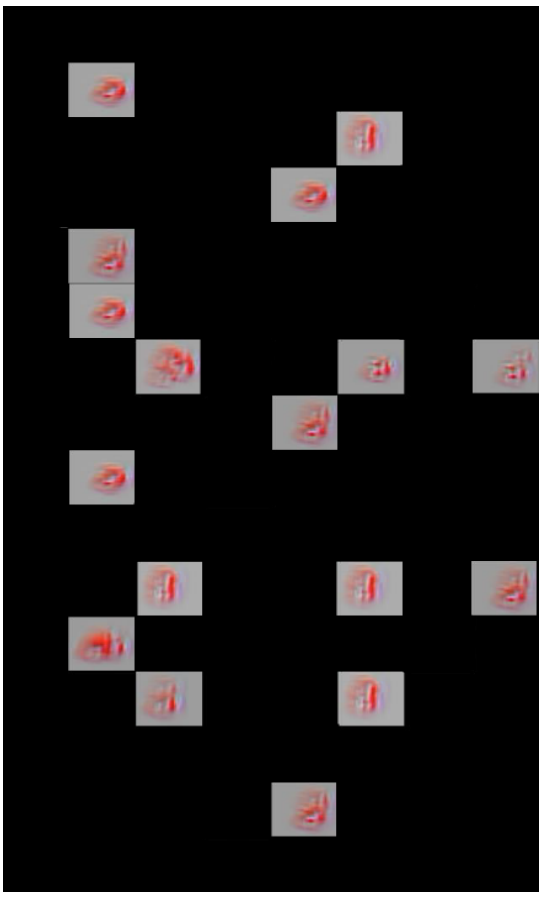

Fig. 9. Filter activation visualization for DCGAN-generated image

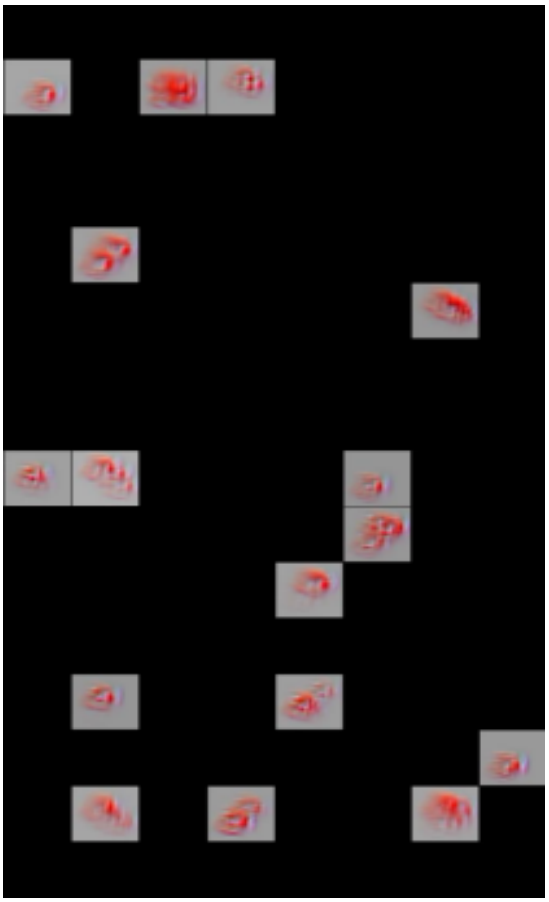

Fig. 10. Filter activation visualization for PGGAN-generated image

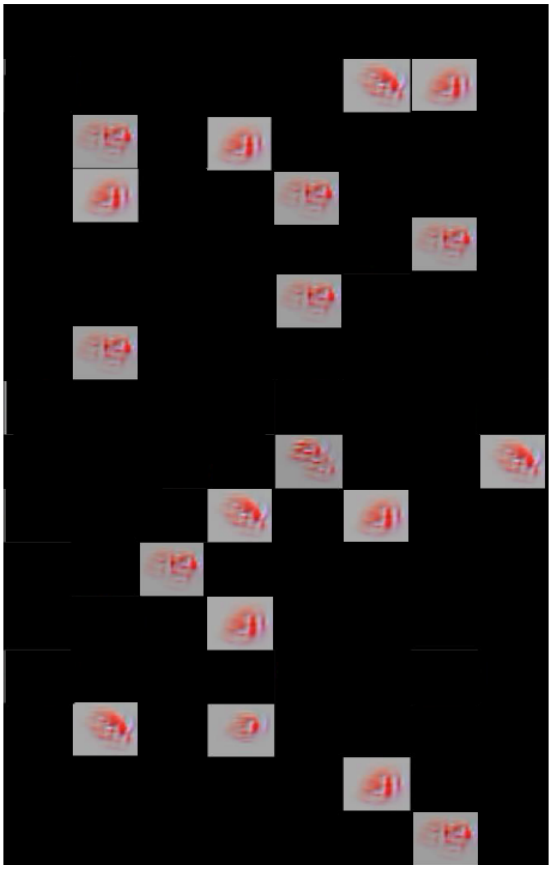

Fig. 11. Filter activation visualization for WGAN-CP-generated image

5.5. Latent Space Visualization

To visualize the latent dimension of the discriminative features, which are also known as CFFs, we implemented a technique for latent space visualization. First, we extracted tensors from the outputs of the last three dense blocks in our CFFN and passed them through separate fully connected layers, resulting in three normalized tensors of equal length. These tensors were concatenated to create a single tensor of length 384, and the loss was computed between the tensors of our (real, real) and (fake, fake) pairs. The aim was to visualize these latent tensors between all images in our dataset. We applied t-Distributed Stochastic Neighbor Embedding (t-SNE) according to [8], to map the features into a 2D space. One thing that we particularly noticed was that the outputs generated by t-SNE are highly dependent on the inputs provided.

Therefore, we selected multiple perplexity factors (number of neighbors), to ensure that the resulting space is well-separated. We observed that using factor as 10, as shown in Figure 12, nearly all features to be mapped were onto a single linear line, which is most likely due to the way dissimilar points are penalized using different numbers of neighbors while computing similarities in the 2D space using the t-distribution.

We found that a perplexity factor of 30 yielded the most promising results, as shown in Figure 13. Additionally, the outcome of training the CFFN model is reflected in the high degree of separability between the real and fake data points, represented by the labels 1 and 0, respectively.

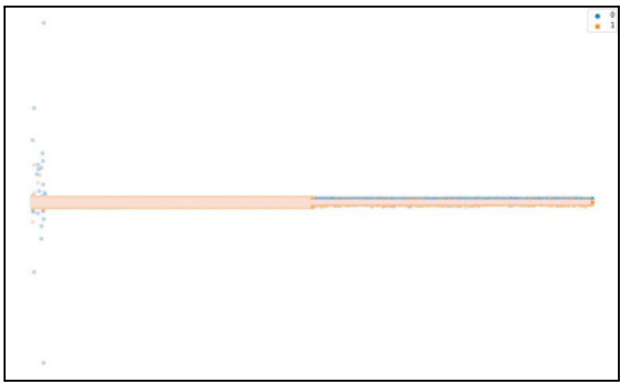

Fig. 12. Latent Space Visualization (CFFN) by t-SNE, perplexity factor = 10

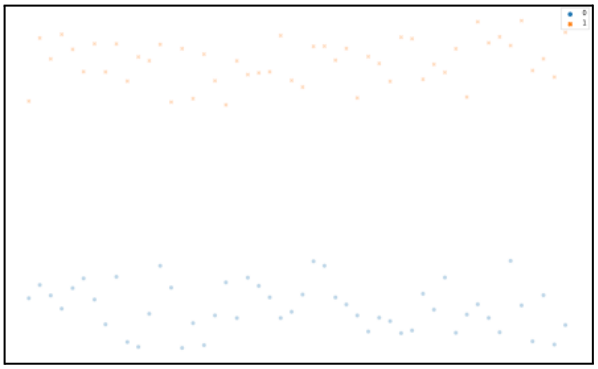

Fig. 13. Latent Space Visualization (CFFN) by t-SNE, perplexity factor = 30

5.6. Quantitative Evaluation of GAN-Generated Images

We conducted an evaluation of the performance of the network using the GAN dataset. We utilized two metrics, namely Fréchet Inception Distance (FID) and Inception Score (IS), to assess the quality of the generated fake images.

According to [9], “Fréchet inception distance (FID) is a metric used to assess the quality of images created by a GAN. Unlike the earlier inception score (IS), which evaluates only the distribution of generated images, the FID compares the distribution of generated images with the distribution of a set of real images ("ground truth").” Hence, the drawback of the Inception Score is that the statistics of real world samples are not used and compared to the statistics of synthetic samples.

FID scores were calculated using methods described in [9], while the computation of IS scores was based on methods described in [10], as displayed in Table 3.

GAN Type	FID Value	Inception Score
DCGAN	0.0514	2.9298
WGAN-CP	0.038078	2.4987
PGGAN	0.03945	2.5081

Table 3. FID and IS evaluation scores for quantitative evaluation of GANs

The FID score computed for each dataset was found to be low, indicating high similarity between the fake images generated by our GANs and the real images. The FID metric quantifies the distance between the features of the fake images and the features of the real images using the Inception Network. On the other hand, the IS scores obtained were

relatively low as compared to the scores achieved by state-of-the-art GAN models, where a high score would be around 10. These scores provide us with a better understanding of the quality of the data used for training.

6. Conclusions

The project proposed a method for detecting fake face and general images generated by the state-of-the-art GANs, namely, DCGAN, WGAN, and PGGAN by using a fake feature network-based pairwise learning. It effectively learned the middle- and high-level discriminative features by aggregating cross-layer feature representations.
Additionally, the pairwise learning strategy facilitated the fake feature learning, which enhanced the trained fake image detector's ability to detect fake images generated by new GANs not included in the training phase. Visualizing the learning capabilities of the CFFN network through reconstructing the final convolutional layer filter activations helped confirm its ability to learn discriminative features distinguishing real and fake images.
This improvement was also evident in its ability to identify the unique and discriminative features present in real images, resulting in more accurate identification of fake and real images. Therefore, our findings suggest that our model can be effective in detecting fake face images generated from different GAN architectures and has the potential to improve its performance with the inclusion of more diverse datasets.

7. Future Scope

The proposed deep fake face detector has demonstrated the ability to detect fake face using deep neural networks. However, there are some limitations that can be addressed in future research. One such limitation is the potential failure of the proposed CFFs to detect fake features of new generators that are significantly different from those used in the training phase. In such a situation, the detector needs to be retrained to adapt to the new generator's features. This can be done by setting up additional datasets for testing our model using available image generation models with different architectures. By comparing the results of these experiments, we can gain insights into how our model performs under different conditions and identify areas for improvement.
Another limitation of the proposed method is the difficulty in collecting training samples due to the technical details of some fake image generators not being publicly available. To overcome this limitation, a few-shot learning policy should be employed to learn the CFF from a small-scale training set. Additionally, future research could focus on developing more efficient methods for collecting and labeling training samples to improve the performance of the proposed method.
Furthermore, the proposed method can be extended to other image manipulation detection tasks, such as deepfakes and image splicing detection. Quantitative evaluation using more methods of the GAN model is another approach to follow. This can be achieved by using various methods such as modified-Inception Score (m-IS), Maximum Mean Discrepancy (MMD), and Classifier Two-sample Tests (C2ST) listed in [7], to get a better understanding of the quality of our training data and the accuracy of our model based on different-rated fake images.
Fake video detection is also an important issue that can be addressed in future work.
In conclusion, there is still much room for improvement and future research in the field of fake image detection, and the proposed method provides a promising starting point for further investigations.

8. References

[1] C.-C. Hsu, Y.-X. Zhuang, and C.-Y. Lee, “Deep fake image detection based on pairwise learning,” Applied Sciences, vol. 10, no. 1, p. 370, 2020.

[2] L. Ziwei, L. Ping, W. Xiaogang, T. Xiaoou, “Deep Learning Face Attributes in the Wild,” Proceedings of International Conference on Computer Vision (ICCV), Dec. 2015.
[3] K. Can, Jan. 2021, “DCGAN (Version 1.0),” [https://github.com/cankocagil/DCGAN].url]
[4] Y. Joona, June 2022, “WGAN-CP with CelebA and LSUN dataset (Version 4.0),” [https://www.kaggle.com/code/joonasyoon/wgan-cp-with-celeba-and-lsun-dataset/notebook.url]
[5] H. Fair, “Progressive Growing of GANs (PGAN),” [https://pytorch.org/hub/facebookresearch_pytorch-gan-zoo_pgan.url]
[6] R. Alec, M. Luke, C. Soumith, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks,” https://arxiv.org/abs/1511.06434, Jan. 2016.
[7] J. Brownlee, “How to evaluate generative Adversarial Networks,” https://machinelearningmastery.com/how-to-evaluate-generative-adversarial-networks.
[8] “Sklearn.manifold.TSNE,” scikit. [Online]. Available: https://scikitlearn.org/stable/modules/generated/sklearn.manifold.TSNE.html
[9] J. Brownlee, “How to Implement the Frechet Inception Distance (FID) for Evaluating GANs,” https://machinelearningmastery.com/how-to-implement-the-frechet-inception-distance-fid-from-scratch/.
[10] “On the evaluation of Generative Adversarial Networks,” https://towardsdatascience.com/on-the-evaluation-of-generative-adversarial-networks-b056ddcdfd3a